$2.00

BEEBLET

Contents:

Editor says	2
Index to advertisers	3
Wordwise help	4
View help	4
December errata	4
6845-cursor help	5
Help wanted	7
Motor on-off	8
Control tv display	9
Adventure note	9
Impossible colours	10
Hall of fame	10
Fun maths - review	12
100 programs - review	13
Cursor reinstated	13
Advanced Graphics - review	14
BCPL	16
Composers + players	22
Classified Ads	22
BBC 52k	25
Advanced User Guide - review	28
GCOL 1,X + 2,X	30
Local meetings...	32

Do Beeblets eat frogs?

THE BBC MICROCOMPUTER USERS GROUP OF NZ INC

NEWSLETTER

VOLUME 2 NUMBER 1 FEBRUARY 1984

EDITORIAL

Even if this is usually the one section in the newsletter you don't ever look at, please do so this time!!!!!!

1. As editor, the Users Group has been reimbursing me the cost of consumables used in producing this newsletter, at the rate of $15 per issue. I am used to spending up to 100 hours each month on preparing the newsletter plus related correspondence etc, must have a new printer ribbon every second month... In July 1982 I tried to resign, but no-one was willing to undertake the job.

At the December committee meeting, Anton moved that consideration be given to increasing the 1984/5 fees by $10.00; most of which would go towards the Editorship, which may be an incentive to someone. Based on current membership this would be around $2,500-3000 in value. There are two variables in this amount - the number of members, and the effects of inflation etc on our Printer's bill, plus that last year's budget was based on 24 pages and now we are up to 32 pages!

Bruce Wills is prepared to take up the Editorship on this basis - he needs, for instance, a printer better than a Creed teleprinter, and several other bits and pieces to maintain the current standard.

MONEY

The payment of Editor and resultant increase in fees is mooted above - if you object (or think it's too little), now is your chance to write in to say so. Time spent on the Users Group has been at the expense of our household budget -we've bought equipment, software etc that would not have otherwise been bought, and although being a housewife may not be a paid job, it saves money by use of sewing and knitting machines, time to read supermarket specials etc!! The final factor in my repeated resignation is a 'non-maskable interrupt event' in 5 months (=baby #3!). Given how much work is involved, no-one has been prepared to take up the editorship without receiving financial support of at least the magnitude suggested above.

Although annual fees are set by an AGM, that will not be until after April, but renewal notices will need to be sent out before then (you do want an April mag in April don't you?). Renewal notices will be sent out in April to everyone; if at the AGM the amount is voted to be different, people will be reimbursed or billed the difference.

LOCAL GROUP MONEY

If you are a local group who gets us to collect your levy for you, tell us soon how much you want for the 1984/5 year billing.
If you are a local group who would like us to bill members in your area for local meeting costs, also let us know. You also get a copy

of the membership form which has all those useful details such as who is prepared to help locally, or give talks etc, for each member.

DUNEDIN:
A Wellington member, Steve, is moving to Dunedin, and will be contacting some of you re the possibility of setting up local meetings (if you haven't already got them).

MISCELLANEOUS?

MOLYMERX have sent us a copy of their software catalogue. They charge $3.00 for the initial copy, then update it freely when needed. It gives very full descriptions of programs available (within NZ!), so especially for the many members who live some distance from a Beeb dealer, this would be a good way to know more about what you contemplate buying. A dealer with the small NZ market cannot be expected to stock every program available for the Beeb which you can then inspect in the shop before buying, so the detail in this catalogue would be a great help.

ATPL, producers of a sideways ROM board for the Beeb, sent us a blurb sheet about it and their Eprom programmer and Northern's EPROM erasers. Their price list doesn't include overseas carriage. Of the 12 extra ROM sockets it gives you, one can be defined as RAM.

Took a while to click: I read even the adverts in UK Beeb magazines - and staring at me in the face was Wellesley St, Auckland - in the February 'Micro User'! FG Software offering goodies to UK people!! And wanting them.

THANK YOUS:

As retiring Editor, I should do my bit and say Thank You to all who have contributed material, and to all who have co-operated and helped me in various other ways too. My husband especially is looking forward to getting at the computer now!

I hope you will all support Bruce at least as well while he is Editor.

ADVERTISERS INDEX:

Business World	19
FG Software	15
Micrologic	7
Molymerx	27
Pirimai Pharmacy	31

NEXT MONTH:

LOTS!!!!!!
Data Generation
Data Selection
Input
Hangman
reviews of books
" of software...

HELP GIVEN:

WORDWISE:

Thanks to Jean and David who find the Wordwise GF"xxxx" command does work - how?:
Files saved under Wordwise's option 1 (or 3), can be reached from the GF command provided they do not include any control codes (ie any commands sandwiched by an f1 and f2 - they appear in green on a colour screen, whilst text appears in white).

If a file has control codes in it, it should be saved using option 8, then it too will load in and be printed satisfactorily.
Thank you to David, Tim, Lance, Tony, Neville, for your responses to the 6845 Cursor problem - Anton should now be an expert on this aspect of the chip, and elsewhere in the magazine is your explanation for all the others who were puzzled.

VIEW HANGUPS?

A couple of members have had problems getting VIEW to print with their cassette-based system. After printing a few lines, it hangs up. In both cases, the cassette player has not had motor control operating, and although the computer has told the cassette player to wait, it's still processing the first block it got, the tape player kept on winding on, so that when the computer went to get block 2, it got the second half, or block 3, and was unable to use it. Solutions:

1. enable motor control on your tape player if you can (see Classifieds for the cable)
2. buy a tape player that will allow you to enable motor control
3. buy a disc interface and drive!!

DECEMBER ERRATA

Some lulus, weren't there!
p.16 program line 40: VDU23,224,&FF,&EE,&CC,&BB,&BB,&CC,&EE,&FF
line 80: VDU23,224,&00,&FF,&00,&FF,&00,&FF,&00,&FF
p.21 Micro-rhymes came from London, not Sendon!
p.66: first diagram should be:

```
               00011101
         EOR   01001001
         gives 01010100
```

FIG.1 should be:

```
    CHR.(1)          00011000
    EOR  CHR.(2)      ????????  (shifted one bit right)
    GIVES CHR.(3)     00011000
```

There should also have been a listing of a program to generate VDU23 codes included. We'll put it in next month for you.

FOUR LETTER CURSING

Thanks go to Lance, David, Tim, Tony, Neville who responded to 'Cursing the 6845'. The following is a distillation of the contributions - by Anton, who had the initial problem.

The 6845 CRTC chip is a general video control chip that controls horizontal and vertical scanning, scanning types, cursor position, paging, scrolling, and even light pen operation. Busy little chip, isn't it.

Information is available from:

Motorola 6845 Data sheet (from VSI Electronics)
Synertee 6545 Data sheet (from Professional Electronics)
The Advanced User Guide (from Whitcoulls and others)
The BBC Micro Revealed by Ruston (from Others and Whitcoulls)
Beebug Feb'83

The Beeb has to work with the pecularities of the chip. The main problem is that it has 18 control registers, some of which are read only (light pen), some are write only (scan and cursor control), and some are quite normal like ordinary RAM. To access these memories, a nineteenth memory has been added which is a register address. This is linked to register &FF00 of the Beeb. So if you are interested in register 2 say, you put 2 into &FF00, which means you can now read and write into the 6845 register 2 (by putting a number or reading it in &FF01). Now &FF is way up in ROM territory which is supposedly not RAM, and therefore unchangeable. Not true. There are three special input/output pages up there, called FRED, JIM, and SHEILA (UG p.436). You can peek and poke Sheila if you wish, but the results could be unwanted. Best to use the official calls; in this case, not OSBYTE but VDU23:

```
VDU23,0,R#,D,0,0,0,0,0,0
```

or more concisely with the semicolon format,

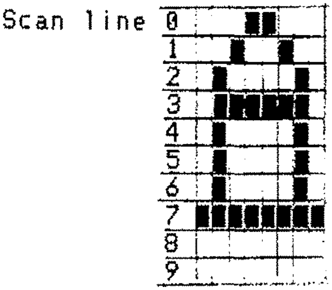
```
VDU23;R#,D,0;0;0;
```

R# is the number of the register of interest, D is a data number that you want to write into R#. You can send them as a pair using the semicolon method. Register 10 with data 32 corresponds to 10,32 or else 8202; (ie 256*32 + 10)

Of direct relevance to the cursor are registers 9, 10, and 11. Characters are organised in rows, with up to 32 scan lines per row. Register 9 sets the number of scan lines. It is increased in text-only modes (3&6) to give a spacing between the lines. Register 10 has 7 bits (numbered 0 to 6 as usual). Bits 0 to 5 set the start of the cursor position. 0011 is a typical value, as that gives the bottom line of a character. The other two combine to affect the cursor display:

Bit 6	Bit 5	Cursor display mode
0	0	Non-blink
0	1	Cursor off
1	0	Blink fast
1	1	Blink slow

Obviously if bit 5 is set and bit 6 isn't (any number between 32 and 63), then the result is no cursor. So my original query about character 32 vs character 33 is simple. Characters are irrelevant. Both numbers turn off the cursor. A fast-blinking, single line cursor should be obtainable by:
VDU23;10,71,0;0;0; (where 71 =1000111 binary)
A thicker cursor is obtained in text only modes because there are two extra scan rows added.



As an example, here is the character 'A', with the cursor at the base.

The advantage of a blinking cursor becomes obvious. One could otherwise confuse it with a 'B' or an '8'. In text only modes, lines 8 and 9 would be part of the cursor.

Register 11 is used to determine the end scan of the cursor. Tony K recommends experimenting, beginning with setting R#,D to 10,0 and then to 11,0. This gives a high non-blinking single line cursor. That is, it underlines the character above the current position.

There are also *FX calls which give the following attributes to the cursor under OS1.0+

*FX154,15	cursor vanishes
*FX154,47	cursor double normal width
*FX154,111	cursor treble normal width
*FX154,235	cursor quadruple normal width

Note that these only work in Modes 4 to 7, with best results in Mode 7.

DISC POWER SUPPLY UNITS:

Where can you get them from, or has anyone the details on how to make them up? A lot of UK mags are suggesting that by the time you run extra ROM or RAM boards etc off the Beeb's own PSU, it will be running pretty hot (if at all), if you try to run single let alone dual drives off it as well.

HELP PLEASE?

Jean has '21 Games for the BBC Micro'. Its Noughts and Crosses program is supposed to draw the 4 grid lines with 4 pairs of commands. However the first pair

```
1090 PLOT 4,450,900
1100 PLOT 1,0,600
```

do not produce a line, only a little dot down in the left hand corner. She can make it draw the correct line by repeating the two commands, as line numbers 1105 and 1106. But WHY?? The two preceding commands are:

```
1070 VDU23;8202;0;0;0;
1080 GCOL 0,3
```

and you are in Mode 5 at this time.

TAPE TO DISC?

Three people have written in wanting to know how to transfer commercial programs from tape to disc. The programs they quote are all Acornsoft.

The August Beebug offers a way, and in the February 'Micro User' is the news item covering Acorn's clobbering Personal Computer World (''65,000 out of court!) who did the same thing. Apparently Acorn are now going to sell disc versions of their software at half price to anyone who sends in the tape (original) to them.

SHUGART FLOPPY DISK DRIVES FOR THE BBC MICRO

Available NOW for the BBC MICRO these drives are the very latest in disk drive technology.

The SHUGART SA465 is a double sided 80 track drive capable of holding 400k using Acorns standard DFS.

PRICES..	SA465 uncased	ONLY $520 tax paid.
	Single drive cable	$ 28
	Single joystick units	$ 33

POSTAGE Drives (and extras) are sent by courier to anywhere in N.Z. for $6.
Add $1 per order for seperate cable and joystick orders.

Micrologic

P.O. BOX 580
MANUREWA
(9) 267-6786 a.h.

MOTOR ON AND OFF via OSBYTE 137

Anton

For those with cassette systems with motor control, it is handy to define +motor0 and *motor1 into the soft keys. Rather than use two keys, a single key can be used that changes the motor status. The following two programs are each capable of doing it. Despite appearances, they really are identical. OSBYTE call 137 is used (U.G.p.433). The status is kept in address &90. When called, the program ensures the contents of &90 are 1 or 0, changes them to 0 or 1, stores a copy of the result back in &90 then calls Osbyte 137. The program assembly has a final RTS to terminate the call. Any key could have been defined this way. Program 1 uses key 4; program 2 uses key 6.

```
 10 REM ****Assemb. Motor on_off
 20
 30 P%=&91
 40 [OPT3
 50 LDA &90
 60 AND #1
 70 EOR #1
 80 STA &90
 90 TAX
100 LDA #137
110 JSR &FFF4
120 RTS
130 ]
140 ?&90=0
150 *KEY4CALL&91|M
```

```
>RUN
0091          OPT3
0091 A5 90    LDA &90
0093 29 01    AND #1
0095 49 01    EOR #1
0097 85 90    STA &90
0099 AA       TAX
009A A9 89    LDA #137
009C 20 F4 FF JSR &FFF4
009F 60       RTS
```

A quicker but obscure program is the following:

```
10 REM****M/c code Motor on-off
20
30 !&90=&2990A500:!&94=&85014901
40 !&98=&89A9AA90:!&9C=&60FFF420
50 *KEY6CALL&91|M
```

Actually, it is program 1 fed in as machine code using plings. The op codes were simply read off the listing produced by program 1, in groups of 4 bytes, and the order reversed since pling puts low byte first.

Either program could be made part of a larger Define Keys program. Locations &90 to 9F are reserved for Econet, which most of us do not have. Should this be wanted for an Econetted computer, change every 9 in program 1 that follows an & to a 7 (eg &91 --- &71)

CONTROLLING YOUR TV's DISPLAY

Neville T. Ch'ch

Many of you 'out there', like myself, are using your tv set with the Beeb rather than go to the expense of a monitor (Wot if, says the Ed, you're like us, with NO tv? How do we keep expense down?) This is fine, except sometimes you will find that the display may be off-centre, too far to the right or left, depending on the particular tv. I have managed to fix this using register 2 of the 6845 CRTC

Register 2 controls the horizontal sync, or displacement of the display from the left-hand side of the screen. Normally it contains these values:

MODES 0 TO 3 - 98
MODES 4 TO 6 - 49
MODE 7 - 51

Try altering these values by a small amount (eg 1 or 2). You will find that decreasing the number will shift your display to the right, whereas increasing the number will shift it to the left.
To write the value 'x' into the second register, use:

```
     VDU23,0,2,x,0,0,0,0,0,0
     OR
     VDU23;2,x,0;0;0;
```

I have found for my tv, that VDU23;2,96,0;0;0; centres the screen in Modes 0-3, VDU23;2,46,0;0;0; in Modes 4-6, and VDU23;2,50,0;0;0; centres Mode 7. However, different tvs may need different values, which you can find by experimenting. (Ed- change the background colour so you can see where the 'screen' is on the screen easier for centring).

For altering the display vertically, it is far easier to use the *TV command on p.23 of the Users Guide. After typing this command the display is altered the next time you press BREAK or change mode.

<u>ADVENTURING NOTE:</u>

Bernard H. (Hamilton) has completed The Castle of Riddles, and will offer help to any who are completely stuck.
Steve W. has got to the ending of Colossal Adventure, and reckons he can help anyone on that one.

Reminder that he also wants to set up a group for Adventuring, and like-minded people should let him know at 61 Disraeli St, Auckland 3.

IMPOSSIBLE COLOURS:

G.C.F.

In addition to the 8 'pure' colours it is possible to add 124 in-between shade fairly easily. To get a pink colour you combine red and white. Well, not quite, you actually use red and white for alternate pixels in a checkerboard pattern. (This is one technique where a colour tv might give a better result than an RGB monitor!)

This can be easily done by drawing dotted lines (PLOT21 - UGp.319). Each line is drawn as 2 dotted lines offset by the width of a pixel (4 in Mode 1, or 8 in Mode 5). To avoid a vertical striping effect, we alternate the order in which we use the colours.

The following program illustrates this technique. Using PROC_M10 which does the actual colour shading (the name has historical reasons - change it to some more meaningful name if you wish). An array of squares is set up then the colours are changed to show how various combinations look against different backgrounds. The program will take about 8 minutes to run through its paces. It may be more useful to change the program to allow the user to determine which combination of colours are displayed. Others may like to try out the flashing colours!

HALL OF FAME:

PUBLISHER:	GAME	HISCORE	SCORER
Acorn	Arcadians	12,640	Brett A.
	Castle of Riddles	**done**	Bernard H.
	Countdown to Doom	127	Tim R.
	Hopper	20,908	Brett A.
	Meteors	15,600	Brett A.
	Monsters	21,200	Athol
	Philosophers Quest	236	Judy
	Planetoid	286,910	Brett A.
	Rocket Raid	15,100	Brett A.
	Snapper	250,000	Susan
		32,660	Sarah (8yrs)
	Super Invaders	19,610	Brett A.
Prog.Pow	Killer Gorilla	91,000	Susan
Level 9	Colossal Adventure	'too busy	playing to note!'
	Dungeon Adventure	330	Anne
	Snowball	550	Gavin

HALL OF FAME

```
   10 REM Impossible Colours
   20 MODE1
   30 VDU23;11,0;0;0;0
   40
   50 REM Set up mixed colour squares
   60 FOR I%=1 TO 3
   70   FOR J%=0 TO 3
   80     IF I%>=J% THEN PROC_M10(I%,J%,320*I%,256*J%,320*I%+200,256*J%+200)
   90     NEXT J%
  100   NEXT I%
  110
  120 REM Now change actual colours
  130 FOR C0% = 0 TO 7
  140   FOR C1% = 0 TO 7
  150     FOR C2% = 0 TO C1%
  160       FOR C3% = 0 TO C2%
  170         IF NOT (C1%=C2% OR C1%=C3% OR C2%=C3%) THEN PROC_change
  180         NEXT C3%
  190       NEXT C2%
  200     NEXT C1%
  210   NEXT C0%
  220 END
  230
  240 DEF PROC_M10(C1%,C2%,LX%,BY%,RX%,TY%)
  250 LOCAL CA%,CB%,T%,Y%
  260 CA%=C1%
  270 CB%=C2%
  280 FOR Y%=BY% TO TY% STEP 4
  290   GCOL 0,CA%
  300   MOVE LX%,Y%
  310   PLOT 21,RX%,Y%
  320   GCOL 0,CB%
  330   PLOT 21,LX%-4,Y%
  340   T%=CA%
  350   CA%=CB%
  360   CB%=T%
  370   NEXT Y%
  380 VDU20
  390 ENDPROC
  400
  410 DEF PROC_change
  420 LOCAL dummy
  430 VDU19,0,C0%;0;
  440 VDU19,1,C1%;0;
  450 VDU19,2,C2%;0;
  460 VDU19,3,C3%;0;
  470 dummy=INKEY(100)
  480 ENDPROC
>
```

Do-It-Yourself Beeblet Kit

LET us know how you see him on our cover

FUN MATHS:

Reviewed by Gio A. Waikouaiti

SQUARE PUZZLES from Scorby Software, Main St, Flixton, Scarborough YO11 3UB. £6 plus £1 p&p

This tape (for Model B) contains four maths games, each a variation on the 'magic square' theme, in which a grid containing number patterns is to be completed. The reward for completing the task, differing in every game, could be the rescuing of a girl by helicopter, ring a bell, or save a parachutist from a nasty fall. The graphics, simple but effective, are accompanied by sound effects that can be turned off when necessary.

The purpose of the games is to improve mental arithmetic (tables) for those aged 7-13 years, and to cater for the ability differences. Each game has a range of difficulties, and sometimes a range of grid sizes. However, in general classroom use I wouldn't attempt to ask children younger than 8 years to cope with it, and believe it would be of most use in a class of 10-11 year olds.

Instructions preceding each game are clear and it took a group of children a very short time to work out how to play. Reading skills would have to be, again, at about the 9-10 year level. The program moves easily from instructions to a friendly scoring system at the end, and if you "give up", the answers are displayed, with explanations in some cases - a good feature. My only gripe is that when figures are displayed in columns, single figures are placed in the tens column which is a little confusing for adding and subtracting (for me anyway).

Comments from the children, even though they were brought back to school during the holidays, were favourable - "It's fun", "You know straight away when you're right or wrong", "Can we use it in achool time?".

My choice of the four games is PATTERN in which logic, number pattern skills and basic facts can be taught more efficiently and easily than I could from a book or blackboard. A library of these sorts of programs would be very valuable in a primary school, although I see that Scorby Soft's catalogue lists only one other for this age group - "Invader Maths" at £4.00 (plus £1 p&p)

(Anyone wanting to find out what else Scorby Soft are offering could send an sae to the Users Group - it'll be cheaper than airmailing UK - Ed)

* * *

100 PROGRAMS

Reviewed by Anton

GORDON, John. 100 Programs for the BBC Microcomputer. pub Prentice-Hall & Acornsoft 1983 209p. pap $26.90

The author is an enthusiast. Anyone who can recommend the Beeb as a kitchen timer has got to be Beeb-crazy! Unfortunately, he is a developing enthusiast who has published a compilation of programs which are largely of mediocre quality, and do not serve as good programming models. Some idea of his methods can be seen in his suggested key definitions for program development: NEW is defined (but should seldom be needed); but BREAK has not been redefined with OLD. PROCedures are rarely used. This is annoying if, say, one wanted to use the mean and standard deviation as part of a larger program. He hasn't provided a simple facility for removal or transferral. He also appears to program first and think afterwards. Quite clumsy methods are used in some programs, and these cause bugs.

The good thing about the book is that there are indeed lots of programs on lots of topics. As a source book of ideas it has merit. Teachers could use it as a "Fix the following program" source. And to historians in future decades, when trying to trace the course of the momentous computer explosion of the '80s, his book will be a source of unedited programs. We'll have wiped ours out with "improved" versions!

(For those of you who wonder if Anton has ever found a book he likes, await next month's mag! - Ed.)

LITTLE HINT

RESTORING THE CURSOR: - Tim W.
While developing a program recently I needed to turn the cursor off for a section of the program, then restore it later. On p.77 of the Users Guide, I learnt that the cursor could be reinstated by a mode change (OS 0,1). However I did not want to clear the screen.

I found the answer in Jeremy Ruston's book "The BBC Micro Revealed" p.23:
To restore the cursor after deleting it by VDU23;8202;0;0;0; use:

```
modes 0 to 6: VDU23;10,103,0;0;0;
mode 7        VDU23;10,114,0;0;0;
```

ADVANCED GRAPHICS

Reviewer: Tim W.

Angell & Jones. Advanced Graphics with the BBC Model B Microcomputer. *pub. McMillan $37.95*

Most users of a BBC micro have more than a passing interest in getting to grips with the excellent graphics capabilities of their machine, whether for the purpose of presenting a lot of information in a more intelligible form, or writing the ultimate arcade game, or constructing 3-D views, or even listing a program sideways in rainbow colours! This book will help you do all of these things, and a lot more besides.

It covers a number of topics that fall under the broad heading "GRAPHICS". There is extensive treatment of a method of drawing 2- and 3-dimensional objects, and the associated problems of perspective and hidden lines in solod objects. This occupies about half the book. Other chapters cover production of character graphics, Mode 7, and diagrams and data graphs. Another chapter details miscellaneous interesting graphics techniques such as scrolling, and changing the print vector to allow a custom-designed printing routine.

Throughout the book the text is supplemented by numerous examples in the form of program listings and screen dumps. Major program listings are: a character generator which enables you to design, edit and save user-defined characters; a series of interactive programs for constructing rather superior graphs and charts easily; and a very large program for designing displays in teletext mode and displaying up to 20 of these pictures sequentially to achieve animation effects. There are also some very useful listings which are unexpected: for example, a disassembler written in machine code, and a routine which enables you to search through a BASIC program to locate and alter specific variables and text strings.

According to the book these programs are available on a pair of cassettes, but I was told that they would not be stocked in NZ. They could be ordered, but at a cost similar to that of the book. At least the listings in the book are very clear, and I haven't found any typographical errors yet! There is also an excellent index, and a separate index of the PROCedures listed in the book.

Now some words of caution. The book is of use to those who have a cassette-based system (like me), but obviously it will be faster to have discs, especially since a lot of *SPOOLing is involved. Also it is an advantage to have OS1.0 plus. The book is not really suitable for beginners, as it assumes a fairly detailed knowledge of Basic, not to mention Assembler. It would pay to be familiar with matrix mathematics if you want to follow the chapters on 2- and 3-dimensional drawing, at least at the first and second readings. Explanations are brief but adequate: the book covers a lot in its 270 pages. If you want a less complex book, you would be better advised to consider such a book as Steve Money's 'BBC micro graphics and sound'. By way of example, Money's book devotes 6 pages to drawing circles; in 'Advanced Graphics' this sort of knowledge is largely assumed.

One last gripe. The final chapter describes the design of a fast-moving arcade game, and gives several interesting machine-code subroutines to whet your appetite. But there is no listing! If you want to get the game, you must buy the tape.

Overall, this is a very interesting book that will be invaluable to anyone wanting to exploit the BBC's graphics capabilities. The review copy was supplied by Whitcoulls; I will not be returning it!
(They will get a small piece of paper though - Ed)

```
FGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFG
FG                                                FG
FG    FG Software PO Box 6662 Auckland            FG
FG                                                FG
FG    Two adventures suitable for the             FG
FG    novice but will provide a challenge to      FG
FG    the experienced adventurer as well          FG
FG                                                FG
FG  -PIRATE ADVENTURE......Seek out the lost      FG
FG    gold of Captain Flint.                      FG
FG  -VAMPIRE ADVENTURE.....Solve the mystery      FG
FG    of the "Count's" castle                     FG
FG      Both $19.95 or $24.95 on disk             FG
FG  -SHUMWARI MENU PLANNER (just arrived!)        FG
FG  -747 FLIGHT SIMULATOR(DOCTOR SOFT)            FG
FG  -GORF 4 different screens of machine          FG
FG    code invaders!                              FG
FG       All $24.95(cassette) $32.95(disk)        FG
FG      Full range of Acorn,IJK,Micro Power       FG
FG      (send SAE for our FREE catalogue)         FG
FG      WE WILL SELL YOUR PROGRAM FOR YOU         FG
FGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFGFG
```

BCPL - AN INTRODUCTION

DUNCAN McE Wgtn

This article is intended to provide an brief introduction to the programming language BCPL. A follow up article will review the BCPL system available for the BBC micro. Before discussing the language, I will briefly describe it's main aims, to give you some idea of the reasons behind its more unusual features.

Language Background.

BCPL was developed in 1967 at London and Cambridge universities, as a language for systems programming. At that time, assembly language was usually required for such tasks, since the high level languages available were not efficient or flexible enough. BCPL is usually a compiled language (a program known as a compiler must analyse the source program, converting it to machine code, before it can be run). This is in contrast to an interpreted language such as BBC Basic, where each source line is converted to machine code as the program is running.

In order to allow the BCPL compiler to generate efficient machine code, the language was deliberately kept small. One way this was done was to only have one simple data type - the bit pattern - but to provide enough primitive operators to manipulate it in any way required. As well as the usual arithmetic and relational operators, BCPL allows bit manipulation and has an indirection operator (like ! in BBC BASIC). It also has an operator for finding the memory address of a variable. These latter two operators combined, allow manipulation of more complex data.

Supporters of BCPL claim that this approach gives a programmer freedom to choose the best way of doing something, rather than having to accept an inappropriate choice by a compiler. They say that BCPL is an ideal combination of a high level language, with assembly language, since it offers many of the advantages of the former, together with the freedom, flexibility, and efficiency of the latter.

Components of a Program.

A BCPL program consists of routines (similar to BBC BASIC PROCs) and functions. The method of declaring these will be illustrated later in an example program. One of the routines in the program must be called 'start' - when you run the program this routine is called automatically.

BCPL Commands.

The simplest BCPL commands are calls to other routines in the program, or assignment statements. A routine is called by using its name, followed by its parameters in parenthesis. If the routine does not have any parameters, empty parentheses are required. The rules concerning the use of parameters are identical to BBC BASIC's.

The general form of an assignment statement is

```
     V1, V2, ... , Vn  :=  E1, E2, ... , En
(n is usually 1)
```

which assigns the the value of expression E1 to variable V1, E2 to V2, etc. As in BASIC an expression may be as simple or complex as required. It may contain variables, constants, any of the operators, and calls to functions (similar to routine calls described above).

BCPL also offers several ways of specifying loops and conditional execution. The general forms of these commands are given below. In each one, E is an expression, which will evaluate to true or false, and C, C1, etc is any single command - including another conditional or looping command. In addition it is a general rule that wherever one command is allowed, a compound command (several commands enclosed in $(and $) section bracket symbols) can be used.

The conditional commands are :

```
IF E THEN C               (if E is true do C)
UNLESS E DO C             (if E is false do C)
TEST E THEN C1 ELSE C2    (if E is true do C1 otherwise do C2)
```

The looping commands are :

```
WHILE E DO C              (Test E.  If true do C and repeat)
UNTIL E DO C              (Same as above, but if E is false)
C REPEATWHILE E           (Same as while, but do C at least once)
C REPEATUNTIL E           (Same as until, but do C at least once)
FOR V = E1 TO E2 BY K DO C  (Same as BASIC FOR-STEP loop)
FOR V = E1 TO E2 DO C     (Same - but step is 1 by default)
```

The K in the 'BY K' clause of the FOR loop must be a constant value.

BCPL also has two commands which can be used to alter the control flow within a loop. BREAK causes control to resume at the first statement after the loop, and LOOP jumps directly to the loop test. These commands are restricted forms of the dreaded GOTO, and so are considered less harmful! BCPL does have a GOTO command, but with such a wide range of structured commands available the situation would have to be desperate for its use to be unavoidable!

BCPL is free-format, which means that all commands can be split over several lines, and indented in any way the programmer wishes, without altering their meaning.

An Example Program.

I have briefly covered the main features of the language, so now I will look at an actual program, to show how everything fits together. The program is a slightly modified BCPL version of the temperature conversion program from page 115 of the BBC Micro Users Guide. The modifications were so I could use more features of BCPL. Note that this program uses integers since BBC BCPL does not support real numbers.

The 'get' command at the beginning of the program reads the file 'libhdr' (library header) from disk into the program when it is compiled. This file contains definitions for many useful library routines, and is used in most BCPL programs.

The first routine or function declared in a program is introduced via the LET keyword, and subsequent ones by AND. To declare a routine the name is followed by the word BE. For a function, the symbol '=' is used.

The '=' in a function header must be followed by an expression. The function 'kelvin.to.celsius' illustrates this. However, it is often the case that a more complex operation is required as shown by the function 'get.scale'. This is achieved via the VALOF command which can be used anywhere an expression is allowed. VALOF is followed by a single command (usually compound) containing one or more RESULTIS clauses. Execution of a RESULTIS makes the following expression the value returned by VALOF.

One command used below (in the routine 'convert') which I did not mention earlier is switchon. BASIC's nearest equivalent is ON ... GOTO. The expression after SWITCHON is evaluated, and the matching CASE is executed. If none of the cases match, but there is a default, it is used, otherwise no commands are executed. The one restriction is that each of the cases must be a constant.

Finally, BCPL provides several standard routines and functions for doing I/O. Those used below are: readn which reads an integer; rdch which reads a single character; writes - for writing strings; writef (formated write) which can write strings, integers in decimal, octal or hexadecimal, etc.

Summary.

I have only been able to present the main features of the BCPL language. If anybody wants to find out more, they can try to get hold

Businessworld
COMPUTERS LIMITED

11 Johnstone Street.
WELLINGTON
Off Featherstone Street

NOW IN STOCK

Disk Upgrades 1.2 ROMs
Pascal ROMs 400 Kb Diskett Drives
View Word Processor
Dot Matrix Printers from only $890.00
Daisy Printers from a mere $1,850.00
And Lots More

BUSINESSWORLD COMPUTERS LTD.

is dedicated to providing full
Wellington based service and maintenance
for all of our customers

Businessworld
COMPUTERS LIMITED

of a copy of the book "BCPL - the language and its compiler" by Martin Richards (one of the languages designers). I believe there is also a book available in England describing BBC BCPL, which may become available here. Finally, there is the user guide which comes with the BBC BCPL system.

```
//  A program to read in temperatures in Kelvin or Fahrenheit scales and
//  print out the coresponding temperature in Celsius.

get "libhdr"           // Get useful system definitions

let start () be
$(
   let temperature, scale = ?, ?      // Declare temperature and scale
                                      // with unknown initial values
   $(

      writes ("Enter temperature and scale : ")
      temperature := readn ()        // Library routine to read integer
      scale       := get.scale ()    // My own routine - returns scale
      convert (temperature, scale)
   $) repeatwhile true
$)

and convert (temp, scale) be
// Do the conversion and write the result or an error message.  The '*n'
// in the write commands is BCPL notation for a new line.  Without it, the
// cursor is left at the end of the line just written (like a prompt).
$(
   let converted = ?

   switchon scale into      // Find out what scale temp is in.
   $(
   case 'C': case 'c':      // Temp already in celsius so no
      converted := temp     // conversion is required.
      endcase               // Exit the switchon.  (without this, the
                            // next case would be executed as well)
   case 'F': case 'f':
      converted := fahr.to.celsius (temp)
      endcase

   case 'K': case 'k':
      converted := kelvin.to.celsius (temp)
      endcase

   default:       // None of the above so write an error message
      writes ("Invalid scale*n")
      return      // No conversion, so return to 'start'.
   $)

   test converted < -273 then       // Impossible to have temp colder
      writes ("Temperature too cold for this universe!*n")
   else
      writef ("%N'%C is equal to %N'C*n", temp, scale, converted)
$)

and get.scale () = valof
// Reads characters, and returns the first non-blank.
$(
   let ch = rdch ()            //  Declare ch and initialise to next
                               //  character in input
   while ch = ' ' do           //  Read characters until a non blank
      ch := rdch ()

   resultis ch                 // Return the character.
$)

and fahr.to.celsius (temp) = (temp - 32) * 5 / 9

and kelvin.to.celsius (temp) = temp - 273
```

TEXT COMPRESSION

Tim L. of Nelson is toying with writing an adventure type game, and is at the point where text compression methods need to be decided on. He would like to know if anyone has gone through this before, or any other ideas - especially with figures on how much saving you can make by the second method offered:

1. Using 5 bits for each character, a total of 32 characters available, with one for 'Shift' to move to alternate character sets (in the same way as the 5 unit Baudot(?) code for telegraph works) giving a total of 62 characters available. This would lead to a saving of about 37.5% in the memory needed to hold text, compared to one byte per character. The coding/decoding routine would also take some space.

2. replace common words, combinations of letters etc with a shorter code - this seems to be the approach Acorn used in Philosophers Quest. The savings with this method I can't estimate.

NB: Micro User has started an Adventure section from January on, which may also be looking at designing your own. The February Acorn User also has several competent articles on game designing and text compression.

COMPOSERS AND PLAYERS:

Neil B. Mt Roskill

I have been compiling a collection of recorder duets, using the Beeb as a competent (if not virtuoso) player for the lower part. I have developed this program for practising and learning purposes. It sets up data files for each tune, or allows you to load in a particular file (tune). Input is easily achieved from printed music. If you know the basics of reading music, this can be done very easily and quickly; if you don't, it's fun to try!

When entering a new tune, it is advisable to play back your efforts every few bars when entering a tune, to correct any mistakes. Playback speed is widely variable, and uses the Beeb's internal clock to control note duration rather than the coarser 1/20th second increments used in the normal SOUND command. This allows you to input note values other than the simple basics (eg triplets). Another playback feature is pitch control, enabling you to make 1/4 semi-tone adjustments to tune the Beeb to your instrument, or even change key.

The tune is repeated by pressing RETURN. By pressing SPACE while the music is playing, you return to the menu, allowing a restart or further changes to speed, pitch, envelope etc.

In the listing, I have called PROCsave (saving a tune as a data file) twice - this is for those of us with 0.1 OS and its cassette bug. If neither copy is satisfactory, type PROCsave and RETURN until successful.

Does anyone have any envelopes that adequately represent musical instruments? Envelopes 2, 3, & 4 are rather spurious!

££££££ 𝄞𝄞𝄞𝄞 ££££££

CLASSIFIEDS & DEALS:

FORTH PACKAGE: Surplus to requirements. Book plus disc version on tape, of Forth, plus instructions to transfer to disc.
$25.00 from David Jones, 44 Falcon St, Kaikorai, Dunedin.

Cassette leads to allow motor control (ie 7-pin DIN to 3 jacks): $5.00 plus 50c p&p. ph. Kevin Wgtn 845-243 or write c/- Box 9592 Wgtn.
TAPES: C-10s, boxed, leaderless - available from the User group in multiples of 20 $24.00 plus $2.00 p&p.

TELEPRINTERS: Still available via the Users Group, $10 (plus carriage outside Wgtn!)

Teleprinter paper - anyone can buy it from P.O. Stores!

```
 10 REM::COMPOSER::N.E.BAKER 8/83
 20 MODE7
 30 ENVELOPE 1,3,0,0,0,0,0,0,121,-10,-5,-2,126,126
 40 ENVELOPE 2,1,0,0,0,0,0,0,2,0,-10,-5,126,0
 50 ENVELOPE 3,8,1,-1,1,1,1,1,121,-10,-5,-2,126,126
 60 ENVELOPE 4,1,-2,-2,-1,5,5,5,0,-1,-1,-1,126,126
 70 DIM P(500),L(500)
 80 CLS:PRINT'''"This program enables you to compose"'"single part music, play
it, FILE it"'"and retrieve it."
 90 INPUT''"Name your TUNE ",tune$
100 C%=0:E=-15
110 PROCinstruct
120 X=0
130 PROCmenu
140 PRINT "Do you want to save the tune Y/N? "
150 IF GET$="Y" THEN PROCsave:PROCsave:REM delete second save if OS>0.1
160 END
170
180 DEF PROCcreate
190 CLS:PROCstave
200 VDU 28,0,24,18,0
210 REPEAT
220   X=X+1
230   PROCwrite(X)
240   UNTIL L(X)=0:REM=note of 0 length
250 X=X-1
260 VDU 26
270 ENDPROC
280
290 DEF PROCinstruct
300 CLS:PRINT''"Enter rest as Note Z, Octave 1,"'"followed by its length."
310 PRINT'"Accidentals:"'"Enter G# as G3"'"Enter Bb as BB"
320 PRINT'"Eb belongs to the octave BELOW E natural"
330 PRINT"Tied notes of same pitch should be"'"entered as one note"
340 PRINT'"Note lengths can be fractional"'"e.g. triplet quaver = 1.33"
350 PRINT'"To end, enter any note and octave"'"followed by 0 length"
360 PRINT''"press any key to continue"
370 Z=GET
380 ENDPROC
390
400 DEF PROCstave
410 PRINTTAB(20,13);"1= semiquaver"
420 PRINTTAB(20,14);"2= quaver"
430 PRINTTAB(20,15);"4= crotchet"
440 PRINTTAB(20,16);"8= minim"
450 FOR line=1TO5
460   PRINTTAB(20,line+18);STRING$(17,"_")
470   NEXT
480 PRINTTAB(24,23);CHR$(255);TAB(30,20);"#";CHR$(255)
```

Does a Beeblet look like this?

Well, Hawksbeeb does

```
 490 PRINTTAB(20,23);"1"
 500 PRINTTAB(27,21);"2"
 510 PRINTTAB(34,19);"3"
 520 PRINTTAB(19,18);"Octave- E to D#
(Eb)"
 530 ENDPROC
 540
 550 DEF PROCwrite(X)
 560 REPEAT
 570   PRINT'"Note ";X
 580   INPUTnote$
 590   REPEAT
 600     INPUT"Octave 1,2,3,4 ",oct
 610     UNTIL oct>0 AND oct<5
 620   PROCnote_to_pitch
 630   IF P=0 SOUND1,-15,222,5
 640   UNTIL P>0
 650 P(X)=P+48*(oct-1)
 660 INPUT"Note length ",L(X)
 670 ENDPROC
 680
 690 DEF PROCnote_to_pitch
 700 P=0
 710 IF note$="E" P=20
 720 IF note$="F" P=24
 730 IF note$="F3" OR note$="GB" P=28
 740 IF note$="G" P=32
 750 IF note$="G3" OR note$="AB" P=36
 760 IF note$="A" P=40
 770 IF note$="A3" OR note$="BB" P=44
 780 IF note$="B" P=48
 790 IF note$="C" P=52
 800 IF note$="C3" OR note$="DB" P=56
 810 IF note$="D" P=60
 820 IF note$="D3" OR note$="EB" P=64
 830 IF note$="Z" P=1
 840 ENDPROC
 850
 860 DEF PROCsave
 870 CLS:PRINTTAB(0,16);"Advance tape
 to blank portion and"'"press RETURN";
 880 dummy=GET
 890 E%= OPENOUT tune$
 900 PRINT'"Please wait"
 910 PRINT#E%,X
 920 FOR J=1 TO X
 930   PRINT#E%,P(J),L(J)
 940   NEXT
 950 CLOSE#E%
 960 ENDPROC
```

```
 970
 980 DEF PROCload
 990 PRINTTAB(0,16);"Press any key an
d play tape at"'"correct place for ";t
une$
 1000 Q=GET
 1010 PRINT"Please wait"
 1020 E%=OPENIN tune$
 1030 INPUT#E%,X
 1040 FOR J=1 TO X
 1050   INPUT#E%,P(J),L(J)
 1060   NEXT
 1070 CLOSE#E%
 1080 ENDPROC
 1090
 1100 DEF PROCsetspeed
 1110 INPUT'"Enter crotchets per minut
e"'"RETURN enters 100 "cpm:IF cpm<20 c
pm=100
 1120 ENDPROC
 1130
 1140 DEF PROCmusic
 1150 PROCsetspeed
 1155 wait=INKEY(100):REM gives you ti
me to pick up your instrument
 1160 PRINT'"You can interrupt the mus
ic while"'"playing by pressing the SPA
CE BAR"
 1170 J=1
 1180 WAIT=INKEY(100)
 1190 REPEAT
 1200   K=INKEY(0):REM allows music to
 be interrupted with space  bar
 1210    IF P(J)=1 THEN V=0 ELSE V=E
 1220   T%=1500*L(J)/cpm+TIME
 1230   SOUND1,V,P(J)+C%,-1
 1240   REPEAT UNTIL TIME>=T%
 1250   SOUND&11,0,0,0
 1260   J=J+1
 1270   UNTIL J>X OR K=32
 1280 ENDPROC
 1290
 1300 DEF PROCmenu
 1310 REPEAT
 1320   CLS
 1330   PRINT'"RETURN- Play music"'"1.
 create NEW music"'"2. change overall
pitch"'"3. change envelope"'"4. displa
y data"'"5. edit wrong note"
```

```
 1340   PRINT "6. add more notes"'"7.
list instructions"'"8. load music"'"9.
 save or end"
 1350   INPUT A
 1360   IF A=0 AND X>0 THEN PROCmusic
 1370   IF A=1 THEN PRINT "Are you sur
e Y/N ? ";:IF GET$="Y" THEN X=0:PROCcr
eate
 1380   IF A=2 INPUT'"Enter 1,2,-1,-2
etc."'"RETURN to restore original pitc
h "C%:PROCmusic
 1390   IF A=3 INPUT'"Enter envelope 1
,2,3,4 "'"RETURN to cancel envelopes "
E:IF E=0 THEN E=-15
 1400   IF A=3 PROCmusic
 1410   IF A=4 PROCdisplay
 1420   IF A=5 INPUT'"Enter number of
note ",N:PROCwrite(N)
 1430   IF A=6 PROCcreate
 1440   IF A=7 PROCinstruct
 1450   IF A=8 THEN PROCload
 1460   UNTIL A>8
 1470 ENDPROC
 1480
 1490 DEF PROCdisplay
 1500 K=0
 1510 PRINT"Press down any Key"'
 1520 REPEAT
 1530   K=K+1
 1540   PROCconvert
 1550   IF GET PRINT "No.";K;" ";Note$
"  Oct ";octave;"  Length ";L(K)
 1560   UNTIL K>=X
 1570 PRINT'"press RETURN to continue"
 1580 REPEAT UNTIL GET=13
 1590 ENDPROC
 1600
 1610 DEF PROCconvert
 1620 octave=(P(K)+28) DIV 48
 1630 basepitch=P(K)-(48*(octave-1))
 1640 Note$="rest    "
 1650 IF basepitch=20 Note$="E        "
 1660 IF basepitch=24 Note$="F        "
 1670 IF basepitch=28 Note$="F# or Gb"
 1680 IF basepitch=32 Note$="G        "
 1690 IF basepitch=36 Note$="G# or Ab"
 1700 IF basepitch=40 Note$="A        "
 1710 IF basepitch=44 Note$="A# or Bb"
 1720 IF basepitch=48 Note$="B        "
 1730 IF basepitch=52 Note$="C        "
 1740 IF basepitch=56 Note$="C# or Db"
 1750 IF basepitch=60 Note$="D        "
 1760 IF basepitch=64 Note$="D# or Eb"
 1770 ENDPROC
```

BBC MICROCOMPUTER 52K

As discovered by Tony K.   Manurewa

ARIES B20 EXPANSION BOARD from Cambridge Computer Consultants, Cambridge CB1 1BR. £90.00

Have you ever been writing a program to use Mode 0 graphics and been hit with the error message 'No room'. Frustrating, isn't it? When I saw the advert in Acorn User for the Aries B20 board I jumped with delight: here at last was the answer to my prayers. I rang Cambridge Computer Consultants' that night and spoke to a very nice lady by the name of Vicki. She told me most apologetically that they could not accept credit card orders, but if I sent them a cheque for £90.00 to cover the board and postage, they would send me one by return mail. I did, and four weeks later I have a B20 board.

The board measures 15 cm by 8 cm and sits over the 6502 socket inside the Beeb. On the board are 10 2K x 8 static RAM chips (Hitachi 6116-20), 12 assorted LS TTL chips and a forty pin socket. To install the board, all one has to do is carefully remove the 6502 processor from its existing socket. plug it into the 40-pin socket on the B20, then plug the B20 board into the now empty 6502 socket on the Beeb's main board. The ROM supplied with the unit is then plugged into a spare paged ROM socket, and that's it. On turning on the Beeb you are now greeted with 'BBC MICROCOMPUTER 52K'.

How does it work? In the Beeb, video RAM and program RAM are one and the same. If you go into a high resolution screen, you lose program RAM space at an alarming rate. In Mode 0 20K is gobbled up by the screen. What the Aries B20 board does is give you all the RAM back for program use. An additional block of RAM contained on the B20 sits in parallel with the existing RAM. When the system accesses program RAM the control circuitry switches the B20 RAM block into action, and Voila! 28K of RAM in any mode! The nicest thing about all this is that the switch is transparent providing your programs use proper operating system calls and do not access the screen memory directly. For those programs which do access screen memory directly (Acornsoft arcade games etc) the board can be disabled by typing in *XOFF. In Mode 7 of course, you only gain 1K of RAM? Wrong, dear friends: a new (Acorn-approved) function call *FX111 can be used to switch the two blocks of RAM. You may now store data using ?,! and $ into either block and switch blocks at will.. You cannot create large BASIC arrays using this switched RAM.

Worth the money? A definite YES. I can now get on and write my large graphics programs without worrying about that annoying 'No room' message. My Mode 0 RAM size just increased five-fold.

Problems? A couple of minor ones: Cambridge Computer Consultants will not be selling the board through dealers, so you have to order the board direct from England; and the board uses space inside the Beeb usually occupied by my Watfords Electronics 13 ROM card. (I now have a 13 ROM card for sale. Takers? ph. (09) 267-6786). Cambridge are bringing out a 12 ROM card which will be compatible with the B20. Until they do, I have 8 ROMs, two more on the way, and only four sockets. HELP!!!!!!

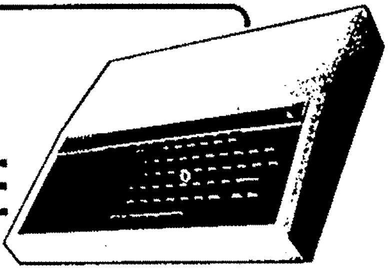
(Jan. Acorn User p.7 also note that Ramamp Computers are producing a 6-ROM card and a RAM card which will both fit under the Keyboard instead of over the main pcb. - would that help? - Ed).

* * *

WANT PEOPLE CLAMOURING TO BABYSIT FOR YOU?
Send Level 9 in UK £9.90 for their latest adventure, Snowball! Wait a couple of weeks for it to arrive. Works for the editor anyway!

Molymerx

BBC SOFTWARE



The MOLYMERX COLOUR SOFTWARE CATALOGUE is ready !

Pages of descriptions of new powerful Application and Utility software for $3.00

and THAT's NOT ALL -

as we publish new software we send you free UPDATES. Our offer is to keep you abreast of NEW products, NEW services and our regular special prices.

Some of the products in the latest edition of our catalogue !

DISK FILE

True Random Access File DATABASE

WAR GAMES

Napoleon Crusaders Emperor

SHUTTLE II

DISK ZAP

MONITOR

Fly COLUMBIA II from space to a Desert landing (if you can !)

Fill in the coupon and return to:

Molymerx

Molymerx Ltd.
P.O. Box 60-152
Titirangi,
AUCKLAND
NEW ZEALAND

(817) 4372

Please send ☐ copies of the Molymerx Software Catalogue.

Please use BLOCK CAPITALS

Name/Company ....

Address ....

..................Zone..................

I enclose $.................@ $4.50 per copy
TRS-80 Catalogue

I enclose $.................@ $3.00 per copy
Colour Catalogue

Australasian distritors for: Aardvark, Acorn, Big Five, C.A.U., A.J. Harding, Hexagon, ICR Futuresoft, Logical Systems, Micro Systems Software, Mumford Micro Systems, Pioneer Software, Understanding, ABC, Powersoft.

Dealer Enquiries Welcome

THE ADVANCED USER GUIDE

Reviewer: Neil G.

Bray, Dickens and Holmes. The Advanced User Guide for the BBC Microcomputer. pub Cambridge Microcomputer Centre 512p. spiral bound

Review copy supplied by Whitcoulls.

Before launching into what is a rave review of this book, I should caution that it is indeed a guide for the ADVANCED user of the BBC computer. For those users (perhaps 20% of you) who intend to program in assembly language, delve into the operating system, or connect exotic hardware to the BBC, this book is indispensable. The other 80% of users have all the information they require in the USER GUIDE supplied with the computer, supplemented by any of the numerous tutorial books on BASIC programming.

The ADVANCED USER GUIDE is a perfect match in appearance and format to the supplied USER GUIDE, and contains almost as many pages. Four main topics are covered - operating system commands, assembly language programming, operating system interfaces, and hardware.

A complete description of all the * commands is contained in the section on operating system commands. Included are several new commands which until now have been documented only in various magazine articles - *HELP, *CODE, *LINE, and *ROM. The section on the *SAVE command also includes a previously undocumented parameter which allows you to specify a reload address for the SAVEd area of memory.

Don't expect to learn how to program in machine language from the section covering this. It is intended as a reference guide only. Once again, it covers topics which have hitherto only appeared in magazines - for instance, the EQUate facility in Level 2 BASIC, which allows space to be reserved for data with predefined values. Fifty-seven pages are taken up with a description of each of the 6502 instructions. Each starts on a new page, which allows for quick reference to the required mnemonic, but does result in a fair amount of blank paper. For each instruction there is a complete listing of the resulting status of registers, and valid address modes and machine cycles taken for each. Surprisingly, the code number used for each instruction type and address mode is not given; this vital information must be obtained from some other book.

The section on operating system interfaces is perhaps the most useful part of the book. Firstly, it contains complete descriptions of the operating system calls, including several calls which had only appeared in magazines, and several that I had never heard of. A particularly interesting one is OSRDRM, which lets you read the value of a byte in any of the paged ROMs. (I used that one to investigate and disassemble WORDWISE). 131 pages are taken up with descriptions of ALL the OSBYTE (*FX) calls, including many whose functions had previously only been guessed at by experimentation.

This section goes on to describe all the OSWORD commands (nothing new here). It then gives a complete list of the vectors used to jump to various standard operating system routines. The advantage of knowing these is that you can intercept a call to a routine by changing the appropriate vector (a two-byte address) to point to your own version of the routine.

Next up is a memory map. As I have pointed out before, where possible you should use the appropriate operating system calls to access operating system parameters rather than relying on their being at a certain address. However, for parameters which aren't accessible through calls, the information provided here will be invaluable. Furthermore, you can use the map to find areas of low memory which are usually safe to store your machine language or variables. For instance, did you know that zero page locations &90 to &9F are allocated to the Econet system? This, of course, means that most people will be able to use these locations for variable storage.

The chapters on events and the RS423/cassette system are excellent, and include several sample programs illustrating the techniques - for instance, reading directly from the cassette port (it does work!). Very full documentation of paged ROMs is given, followed by a chapter on BBC filing systems. The section on cassettes includes the now open secret on the bit that is set for "locked" files.

The hardware section covers much the same ground as the "Expert Guide", but in much more detail, plus it gives information on the Tube and the 1 MHz bus. Also included is a circuit diagram of the BBC.

Overall then, this is an excellent reference guide which combines in once neat package a lot of very useful information which, until now, has only been available in bits and pieces. I heartily recommend it for those users who need the kind of information it provides.

```
10 REM *** GCOL 1,X AND GCOL 2,X DEMONSTRATOR ***
20 REM ***         BY TIM RYAN                ***
30 ON ERROR MODE 0:END
40 MODE 2
50 VDU 23,224,36,126,195,203,211,126,36,66
60
70 REM MAKE ALL THE COLOURS FROM 8 TO 15 DISPLAY WHITE
80
90 FOR colour=8 TO 15
100   VDU 19,colour,7,0,0,0
110   NEXT
120
130 REM DRAW BLOCKS OF COLOUR FOR THE SPACESHIP TO FLY OVER
140
150 FOR horiz_posit=144 TO 1008 STEP 144
160   GCOL 0,horiz_posit DIV 144
170   MOVE horiz_posit,0:MOVE horiz_posit,1000
180   PLOT 85,horiz_posit+128,0:PLOT 85,horiz_posit+128,1000
190   MOVE horiz_posit DIV 9,0:DRAW horiz_posit DIV 9,1000
200   MOVE 1156+horiz_posit DIV 9,0:DRAW 1156+horiz_posit DIV 9,1000
210   NEXT
220
230 REM NOW FLY THE SHIP OVER THE LANDSCAPE USING ORS AND ANDS
240
250 VDU 5
260 ship_old_hpos=1:ship_old_vpos=500
270 horiz_dir=1:ship_horiz_posit=1:vert_dir=0:ship_vert_posit=500
280 REPEAT
290
300   GCOL 2,7 : REM 'AND' WITH HIGH BIT RESET TO REMOVE SHIP
310
320   MOVE ship_old_hpos,ship_old_vpos:VDU 224
330
340   GCOL 1,8 : REM TO PUT UP SHIP IT IS OR'D WITH BACKGROUND
350
360   MOVE ship_horiz_posit,ship_vert_posit:VDU 224
370   IF ship_horiz_posit>1200 OR ship_horiz_posit<1 THEN horiz_dir=-horiz_dir
380   IF RND(10)=10 THEN vert_dir=RND(3)-2
390   IF ship_vert_posit<40 THEN vert_dir=2
400   IF ship_vert_posit>1010 THEN vert_dir=-2
410   ship_old_hpos=ship_horiz_posit:ship_old_vpos=ship_vert_posit
420   ship_horiz_posit=ship_horiz_posit+8*horiz_dir
430   ship_vert_posit=ship_vert_posit+8*vert_dir
440
450   UNTIL INKEY(5)=32
460
470 VDU 4
```

GCOL 1,X AND GCOL 2,X:

TIM R.

The following program is an experiment to allow a foreground object to be animated across a coloured background without either the foreground or background changing colour as the foreground object passes by. I wrote it after seeing the experiments done in the December issue of the User mag where the GCOL 3,X statement was used. To me the most annoying feature of the XORing technique is that the foreground object changes colour as it passes over the varying landscape. This technique avoids the problem but as it is, limits the user to one foreground colour.

However the book 'Advanced Programming Techniques For The BBC Micro' by Jim McGregor and Alan Watt gives a lot more information on this subject including a way to have five separate 'foregrounds' that do not interact with each other. Each plane may be independantly filled and cleared without affecting any of the others' planes, either in front or behind. This gives an amazing effect and is really easy to control. For more on this I can really recommend the book mentioned above. The program runs as is, and may be developed as you like. To stop the animation, simply hit the <SPACE BAR>. Have fun!

Can we help you with items for your B.B.C.

SOFTWARE now in stock:-
Most of the BBCSOFT and ACORNSOFT ranges
The full I.J.K. range
MOLYMERX JUMBO and SPACE SHUTTLE
For serious applications consider the GEMINI range
BEEBCALC BEEBPLOT DATABASE MAILING LIST
STOCK CONTROL HOME ACCOUNTS EASILEDGER
also CASHBOOK and FINAL ACCOUNTS
Just Arrived is the impressive AB2 Designs Package

HARDWARE Now Available:-
Disc Drives 80 track Double Sided (400K) $800
(these can be supplied with 40 track switch)
Printer $750
Acorn joysticks pair $55
Almost Completed 16 Key programmable Keypad

For further information contact:

Bob Brady at Pirimai Pharmacy Ltd.
Pirimai Plaza phone 439016
NAPIER

USER GROUP MEETINGS

AUCKLAND:-meets 2nd Wednesday of the month, 8.00pm, VHF clubrooms, Hazel Ave, Mt Roskill. ph. Dave 770-630 x 518(bus) or Kerry 695-355

HAMILTON:-last Wednesday of the month 5pm Waikato Tech B-block staffroom. Ph. Peter, Hamilton 393-990 or Alison, Morrinsville 6695.

TAURANGA:-every second Wednesday. ph. Chris & Jane 65-076.

HAWKES BAY:-Alternate months Hastings & Napier. Contact Kendall Napier 435-624, Bob Taradale 446-955, Mitch Hastings 778-235.

TAUPO:-Contact David 84-215.

WELLINGTON:-meets LAST Thursday of the month, 7.30pm, Correspondence School Staffroom (1st floor), Portland Cres, Thorndon. ph. Anton 286-289 or Warren 787-005.

CHRISTCHURCH:-fortnightly workshops, Tuesdays 7.00pm at Hagley High School. ph. Michael 582-267.

OTHER CENTRES:- let us know to publish details here.

THE NEWSLETTER:

CONTRIBUTIONS:- most welcome. Listings should be sent in on tape or disc. (Tape - 2 copies, one at 300 baud please) A 35c stamp gets your tape / disc returned too. Please include written explanation of listing; text-only material welcome too.

DEADLINES:- For a particular issue, the last day for material is the last day of the month prior to the month of publication.

ADVERTISING:- Rates are $20 per half page, deadline as above.

Copyright (c) THE BBC MICROCOMPUTER USERS GROUP OF NZ INC.
Published monthly except January and mailed to financial members.

MEMBERSHIP:

Membership of the users group is on payment of an annual subscription from April to March. For the 1983/4 year it is: $20.00, plus a $5.00 joining fee for new members, plus $5.00 for Auckland & Wellington members as a local Branch levy, which entitles attendance at meetings as detailed above. Renewing members get all back issues of the current year; New members get ALL back issues back to v.1 #1 Jan '83.

THE BBC MICROCOMPUTER USERS GROUP OF NZ INC
P.O.Box 9592
Wellington

Reproduced by ROBERTS PRINTING Co. Ltd, Wgtn.